[redacted]

UIUCDCS-R-73-580

USE AND DESCRIPTION OF THE CALCOMP PROGRAM TO DRAW NETWORKS OF LOGIC GATES

by

JAY CULLINEY

June, 1973

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

UIUCDCS-R-73-580

USE AND DESCRIPTION OF THE CALCOMP PROGRAM TO DRAW NETWORKS OF LOGIC GATES

BY

JAY CULLINEY

June, 1973

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

* Programming work was supported by the National Science Foundation under Grant No. NSF GJ-503.

TABLE OF CONTENTS

Introduction:

The program described in this reference manual was developed as a means to quickly obtain a reasonably precise drawing of a logical network, given a description of that network.

Due to the difficulties involved, the program does not guarantee the most "aesthetically pleasing" arrangement of gates and interconnection lines (e.g., crossovers are not minimized) in each network diagram. However, it is still felt that this program produces network diagrams which are reasonable in appearance (e.g., the method employed for routing interconnections possesses at least some degree of sophistication) and acceptable for general purposes.

Program Size: The program consists of three subroutines. When these are compiled with the Fortran H (opt 2) compiler, the resultant program occupies about 80K bytes of memory. The source deck consists of about 750 cards while the object deck (Fortran H, opt 2) contains about 320 cards.

General Description: Given the following information for each gate in a network, this program positions the gates, labels the gates, and determines interconnection paths among gates such that the paths do not overlap with each other or with gate symbols. Provision is made for several lines of information which may accompany each network. (The program listing appears in Appendix B.)

1) Gate Type--e.g., AND, OR, NOR.
2) Gate Level--this is defined for a gate i as follows: the number of gates in the longest path through the network from gate i to an output gate.

3) Gate Inputs--i.e., a list of gates and external variables which feed this gate.

Limitations:

1) Gate types are restricted to: AND, OR, NOR, Wired-AND, Wired-OR, NAND, Exclusive-OR, and Exclusive-NOR
2) Maximum number of gates--20
3) Maximum number of levels--unrestricted
4) Maximum number of external variables--9
5) Maximum number of gates specified per level--10
6) Maximum fan-in--7
7) Maximum fan-out--unrestricted
8) Maximum amount of information allowed with each network--7 lines of 50 characters each

Restrictions (2), (3), (4) above are imposed only by the size of arrays in the program, and hence, are relatively easy to alter. Restriction (5) can probably be increased with only a slight programming change. We should note here that although up to 10 gates may be specified per level, only up to 5 gates are actually drawn in the same level (the other gates specified as being in that level are actually drawn one level of gates higher). This is due to the fact that the width of plotting allowed on usual Calcomp paper is restricted to 10 inches. After allotting space for 7 lines of information, there is only enough space remaining for 5 gates per level. By eliminating the information block, or by moving it elsewhere, we can achieve 6 or maybe even 7 gates per level. If higher limits are desired, one can go to wider Calcomp paper. But any change in the maximum number of gates drawn per level may require moderately extensive program changes. Restriction (6) is

due to the fact that there are only seven positions on each gate where we may attach inputs. By making special provisions, we might increase this to 9 or 11 inputs, but such changes would probably be of moderate difficulty. Restriction (8) would be fairly easy to modify (increasing the number of characters per line to 80 is especially simple) if we do not mind using more Calcomp paper, but it is unwise to write more information than is necessary since the amount of Calcomp time consumed in plotting characters is quite significant.

Exceeding Limitations: If any of the 8 limitations above are exceeded, the following actions are taken by the program, respectively:

1) Gate type OR will be assigned to any gate not specified as one of the 8 allowed types.
2) The program does not check this limit, so if it is exceeded, memory overwriting will occur.
3) Specifying more than 20 levels (or specifying a level greater than 21) will result in memory overwriting.
4) Exceeding this limit will also cause overwriting of memory.
5) The exact consequences of exceeding this limit are not known. Probably either memory will be overwritten, a pen fault will occur, or perhaps both. See Fig. A-3 in Appendix A for the results of specifying more than 5 (but less than 10) gates per level.
6) The inputs to a particular gate are written sequentially on a data card. If more than 7 inputs are listed, only the first 7 are accepted, the rest ignored, and a message is written by the Calcomp plotter: "fan-in overflow, inputs beyond 7 ignored". (See the example in Fig. A-5 of Appendix A.) The plot con-

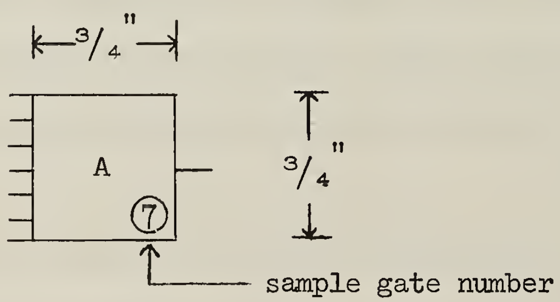
tinues normally so that the user may still obtain a complete network diagram by adding the missing connections later by hand (but in some cases, the error message may obscure a part of the network).

8) If 8 or more lines of information are specified, the 8th line will be omitted, replaced by the warning message: "error--too many heading cards". Any other lines are written in a space to the right of the first information block. If the number of lines in this second block of information again exceeds 7, the whole process is repeated. See the example in Fig. A-4 of Appendix A.

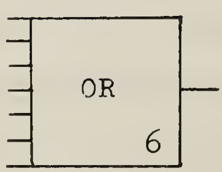
Gate Symbols

The following symbols are used in the networks drawn by this program (the symbols used for AND, OR, NOR, NAND, and Exclusive-OR are the "Uniform Shapes" of the American Standard Graphic Symbols). Also, for each gate, input and output lines are shown. See Fig. A-1 in Appendix A for examples of these different gate symbols as they are actually plotted.

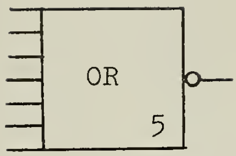
AND Gate:



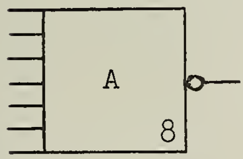
OR Gate:



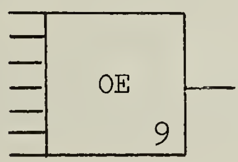
NOR Gate:



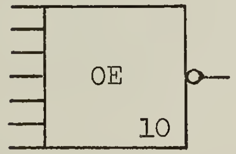
NAND Gate:



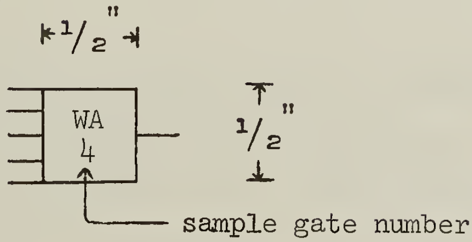
Exclusive-OR Gate:



Exclusive-NOR Gate:

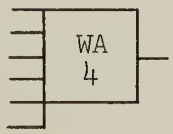


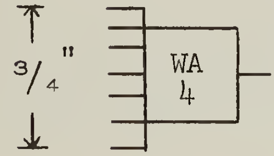
Wired-AND Gate:



(for 5 input case)

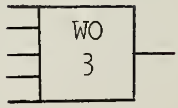
(for 6 and 7 input cases, we have, respectively:





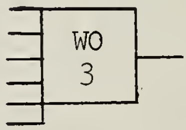
)

Wired-OR Gate:

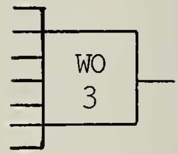


(for 5 input case)

(for 6 and 7 input cases, we have, respectively:

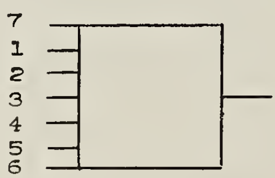


and

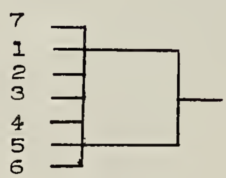


)

For each gate type there is a maximum of 7 inputs. Input terminals of a gate are assigned to input lines in the following order (there exist some rare cases when this order is slightly altered by the program to avoid superimposing interconnections):

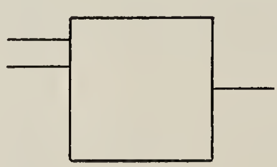


For OR, AND, NOR, NAND, Exclusive-OR, Exclusive-NOR Gates



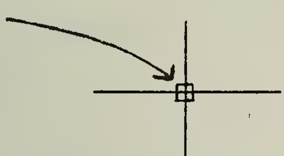
For Wired-AND, Wired-OR Gates

So a gate with k inputs will have those input lines attached to input terminals 1 through k. For example, a gate with 2 inputs would be:



The (possible) nine external variables and their complements are designated: A, B, C, D, E, F, G, H, I, $\overline{A}$, $\overline{B}$, $\overline{C}$, $\overline{D}$, $\overline{E}$, $\overline{F}$, $\overline{G}$, $\overline{H}$, $\overline{I}$, in the output diagram.

When two gate interconnections intersect, a small square at the intersection will indicate the existence of a connection between the two lines (e.g.,



). The absence of such a square will indicate that the two lines are not connected.

Preparation of Input Cards

Of course, there are many possibilities for combinations of JCL cards. In Fig. 1 is a typical deck setup for the Calcomp program (for the current system used at D.C.L. (June, 1971)). There are two things of note here. First is the inclusion of the parameter "CALCOMP=YES" on the ID card. Second is the "// EXEC CALCOMP" card (which physically consists of two cards in this example, i.e., "REGION=232K" is a continuation of the "EXEC CALCOMP" card). Two parameters must be specified on this card: Maximum length (in inches) of paper to be used; Maximum plotter time (in the form h.mm.ss where h is number of hours, mm is number of minutes, and ss is number of seconds). The example in Fig. 1 specifies a maximum of 75 inches of paper or 5 minutes of plotter time.

The block of cards labeled "Data Cards" actually consists of one or more consecutive network descriptions. Each network description consists of four parts:

1) First come the heading cards. On these cards we may write any information that is desired to be output along with the network

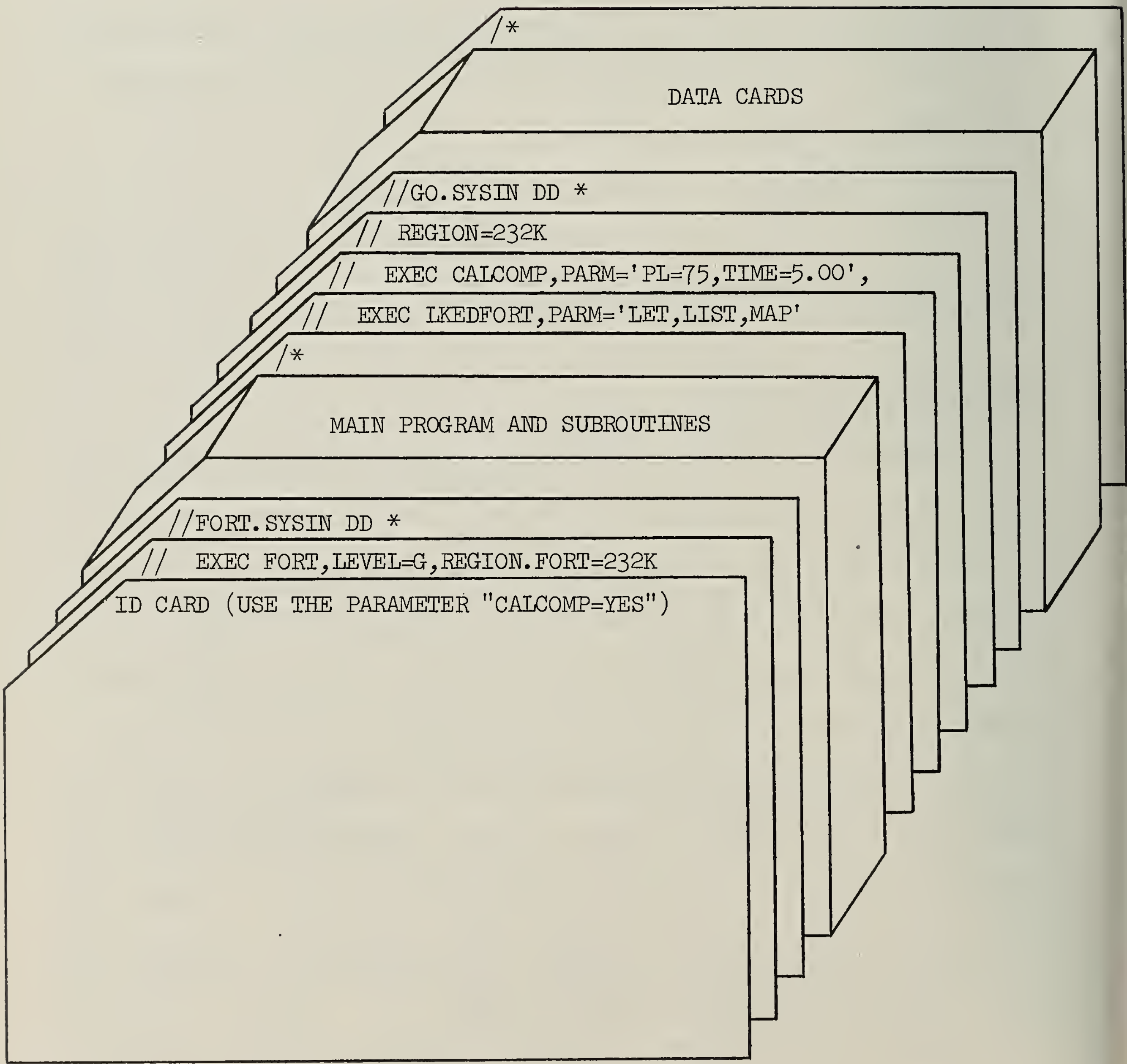

/*
DATA CARDS
//GO.SYSIN DD *
// REGION=232K
// EXEC CALCOMP,PARM='PL=75,TIME=5.00',
// EXEC LKEDFORT,PARM='LET,LIST,MAP'
/*
MAIN PROGRAM AND SUBROUTINES
//FORT.SYSIN DD *
// EXEC FORT,LEVEL=G,REGION.FORT=232K
ID CARD (USE THE PARAMETER "CALCOMP=YES")

Fig. 1

diagram. There may be a maximum of 7 such cards and only columns 1-50 of each card may be used.

2) Next is a card which signals the end of the heading cards. In columns 13-16 must be the characters "b..b" (where b=blank). Anything may be in the other columns. Naturally, such a combination in columns 13-16 must not appear in any of the heading cards.

3) Next is a group of cards, one card for each gate in the network. These <u>must</u> be arranged such that the card for Gate 1 (the first output gate must be number 1) comes first, the card for gate 2 comes second, etc. In columns <u>5-6</u> of each card, we may write two symbols which will be written beside the output line of the corresponding gate (this provision is to enable the user to label outputs of a multi-output network). Normally, for a single output network, this field is left blank. In columns <u>12-13</u>, write the gate number (right justified). In columns <u>17-20</u> write the type of this gate; the following forms are currently accepted (b=blank):

> "bNOR" for NOR; "bAND" for AND; "(AND" for AND; "WORb" for Wired-OR; "WAND" for Wired-AND; "ORbb", "bORb", "bbOR" for OR; "NAND" for NAND; "bXOR" for Exclusive-OR; "XNOR" for Exclusive-NOR

(any other forms, or any mistakes in the above forms will result in the default assignment of the gate type OR). In columns <u>27-28</u>, write the level number (right justified) of this gate. Level numbers need not be assigned in accordance with the definition (of gate level) given earlier. Actually, the only requirement is that the level number of a given gate must be greater than the

level numbers of all gates fed by that gate. Starting after column 34, we have 11 fields of 4 characters each. In these fields (columns 35-38, 39-42, 43-46, . . .) are listed the inputs to this gate, one input per field. In each field we write either the name of an external variable or the number of a gate which feeds this gate. In either case, the characters used must be right justified in each field. (External variables x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, $\bar{x}_1$, $\bar{x}_2$, $\bar{x}_3$, $\bar{x}_4$, $\bar{x}_5$, $\bar{x}_6$, $\bar{x}_7$, $\bar{x}_8$, $\bar{x}_9$, should be punched on the input cards as, X1, X2, X3, X4, X5, X6, X7, X8, X9, Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, respectively. Alternatively, the characters "U" and "V" may be used in place of "X" and "Y" respectively.) Currently, the program will only use the entries in the first seven fields and give a warning message if there are more than seven non-blank fields. The fields are read left to right; reading stops in the first blank field.

4) Last is a single card with the characters "END" in the first three columns.

Construction of the Network

The program determines which and how many gates are in each level. (The output gate(s), at level 1, is to the right of the network. The other levels are then labeled, from right to left, 2, 3. 4, ...) An imaginary horizontal line, whose vertical height is fixed (from the bottom of the Calcomp paper), serves as a horizontal axis about which the network is centered. Within each level, the gates are positioned symmetrically about this line and ordered such that the smallest numbered gate (in that level) is assigned the "highest" position within that level and the highest

numbered gate (in that level) is assigned the "lowest" position within that level. So, at this point, every gate in the network has its vertical position specified. Horizontal coordinates cannot yet be determined (note that every gate of a given level will have the same horizontal coordinate) since the complexity of the interconnection patterns between each pair of adjacent levels is as yet unknown.

Next, external variables are assigned to the input terminals of the corresponding gates.

Then we go through the gates in order, gate 2, gate 3, . . ., determining the routing for the output connections of each one. For example, for gate i, first all levels containing gates fed by gate i are determined. The right-most level fed by gate i is then selected. We locate the specific gate or gates at this level which are fed by gate i. An input line is created for each of these gates. These horizontal input lines (if there are more than one) are joined by a vertical line. To this vertical line is attached another horizontal line which is fed through the next level of gates to the left. Then we repeat the same process (of creating inputs to gates fed by i) in the next level to the left, treating the line we just fed through to that level (almost) the same as any inputs to gates in this new level. Finally we will reach the level containing gate i and we can connect it to its newly constructed output tree.

Example: Output connections from gate i to other gates in the network are constructed. Assume gate i feeds gates 1, 2, 4, 5, 7, 8.

First we find that gate i feeds levels 1, 2, 3, and 4. Of these, level 1 is the right-most. And gate i feeds gate 1 in level 1.

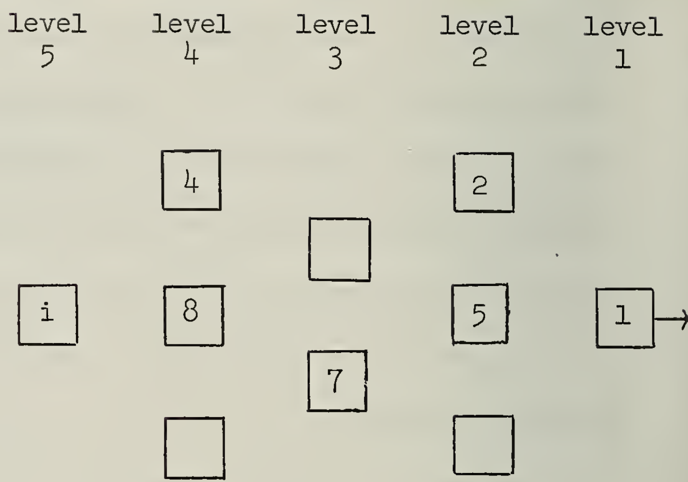


Fig. 2

So we draw an input line to gate 1 (as shown). This line is then fed through level 2 (as shown in Fig. 4).

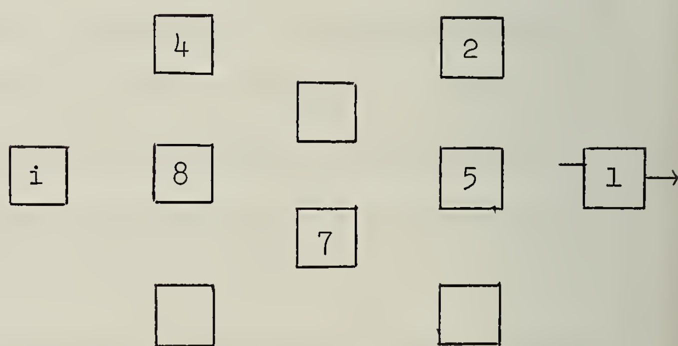


Fig. 3

At level 2 we find two gates which are fed by i, gates 2 and 5. Input lines are created for these gates.

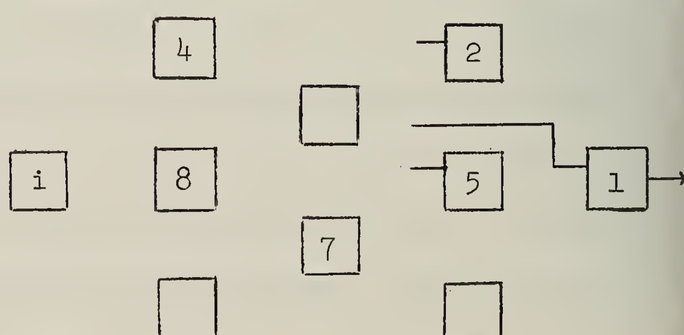


Fig. 4

The three horizontal lines in Fig. 4 are then joined by a vertical line (Fig. 5) and another horizontal line is used to feed through level 3.

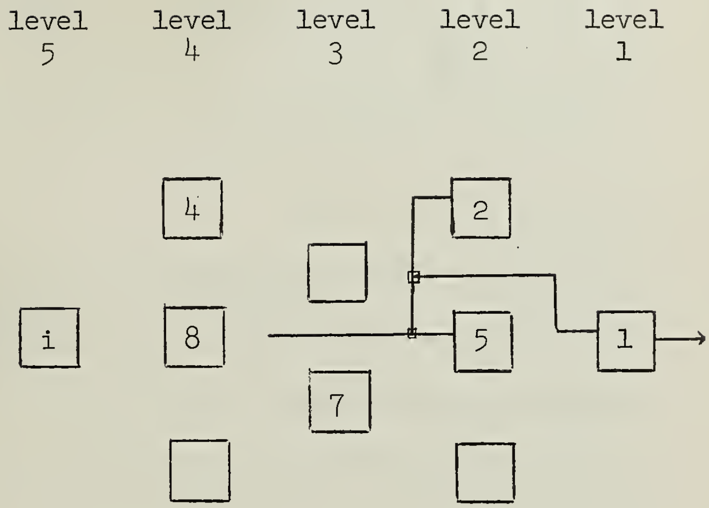


Fig. 5

At level 3 we find that only gate 7 is fed by i. An input line to gate 7 is created and then joined with the line fed through from level 2.

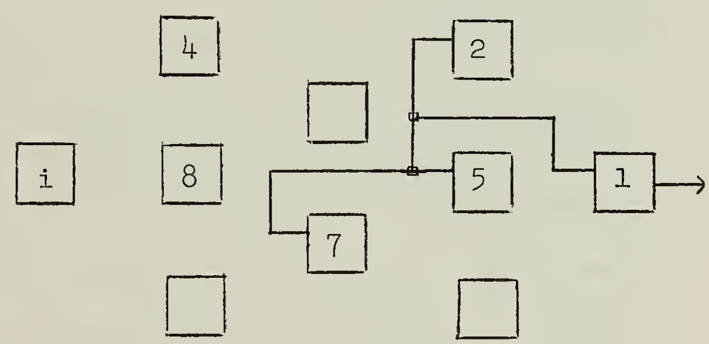


Fig. 6

Again, a horizontal line is fed through the next level to the left. At level 4, input lines are created for gates 4 and 8.

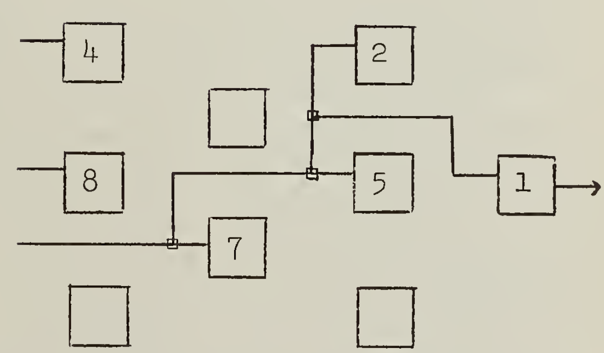


Fig. 7

At level 4 the 3 horizontal lines are joined by a vertical line. It is found that gate i is in the next level, and so a horizontal line is drawn from the output terminal of gate i to its interconnection tree.

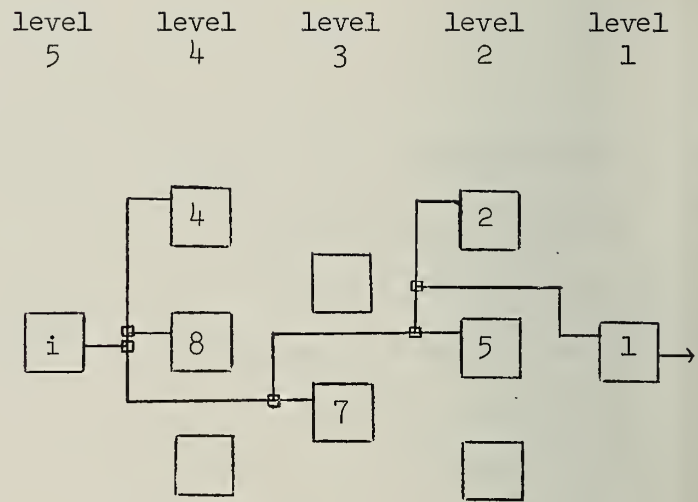


Fig. 8

End of example.

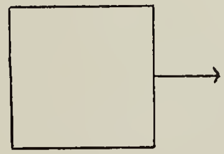
Similar steps must be repeated for i = 2, 3, . . ., R (where R is the number of gates in the network). The routes taken by interconnections are "memorized" as they are determined so that no two interconnections may coincide. Actually the interconnections are not physically "drawn" at the time of determining the routing of the outputs from each gate since the actual x-coordinates of the intersection points cannot be calculated until all interconnection routing is complete and the number of vertical lines necessary between each pair of adjacent levels is known. Instead, we specify the x-coordinates of each intersection point relative to either the previous or succeeding level. This information is stored, and later when the x-coordinates of the levels can be calculated we can easily determine the x-coordinates of all intersections of interconnection lines relative to the origin (which is located at the center of the output level (level 1)). This

information is needed in order to do the actual plotting. Notice that there was no trouble determining the y-coordinates of intersection points. The y-coordinates are known immediately for each intersection, as opposed to the x-coordinates which cannot be determined until all interconnection routing is complete.

It might be mentioned that interconnection lines are kept 1/8 inch apart (except when crossing at right angles). And horizontal lines must be at least 1/8 inch away from the side of a gate, while vertical lines are no closer than 3/8 inch to the right of a gate or 1/2 inch to the left of a gate.

Multiple Output Networks

In assigning gate levels for gates in a multiple output network, all output gates which do not feed other gates should be assigned to level 1. All gates in level 1 will be drawn with an arrow (indicative of an output of the network):



Output gates which do feed other gates should not be assigned to level 1. In order to distinguish the different output functions in a multiple output network, a provision was made (as mentioned earlier) so that we can label the different output functions. For each gate, we can specify a label (of 1 or 2 symbols) which will be written beside the output of that gate (e.g., we can label the outputs of a multiple output network: F1, F2, F3, etc.). Fig. A-2 in Appendix A is an example of a multiple (5) output network with outputs labeled F1, . . ., F5.

<u>Future Modifications</u>--Further modifications of this program <u>may</u> occur in the future. In particular: multiple output gates (NOR-OR, NAND-AND), flipflops, feedback loops, automatic calculation of levels.

Currently, the paths of the output lines for each gate i are determined in the order i=1, 2, 3, . . . , 19, 20. It is possible that assigning interconnection paths for gates in reverse order (i.e., i=20, 19, 18, . . ., 3, 2, 1) would reduce the number of crossovers in the network diagram. But this change is not quite as easy as it appears since some parts of the program depend on the assignment of interconnection paths for gates in ascending order.

<u>Computation Time</u>--For an average size network of 8 gates, we might expect, typically: a computation time of about 1/3 or 1/4 second; a plot time of about 1 minute; a cost of about 40¢.

APPENDIX A

Examples of Networks Plotted by CALCOMP

Table A-1

```
THESE ARE THE DATA CARDS FOR FIG. A-1:



FIG. A-1     THIS EXAMPLE SHOWS THE DIFFERENT
GATE TYPES AND EXTERNAL VARIABLES AVAILABLE.
                ..
     F          1    NØR         1          X1   Y9    2    3    4    5    8
                2    AND         2          X2   Y8    6    7    8
                3   WØR          2          X3   Y7    5    7
                4   WAND         2          X4   Y6    7
                5     ØR         3          X5   Y5    6    7    8
                6   NAND         4          X6   Y4    8
                7    XØR         4          X7   Y3
                8   XNØR         5          X8   X9   Y2   Y1
END
```

FIG. A-1 THIS EXAMPLE SHOWS THE DIFFERENT GATE TYPES AND EXTERNAL VARIABLES AVAILABLE.

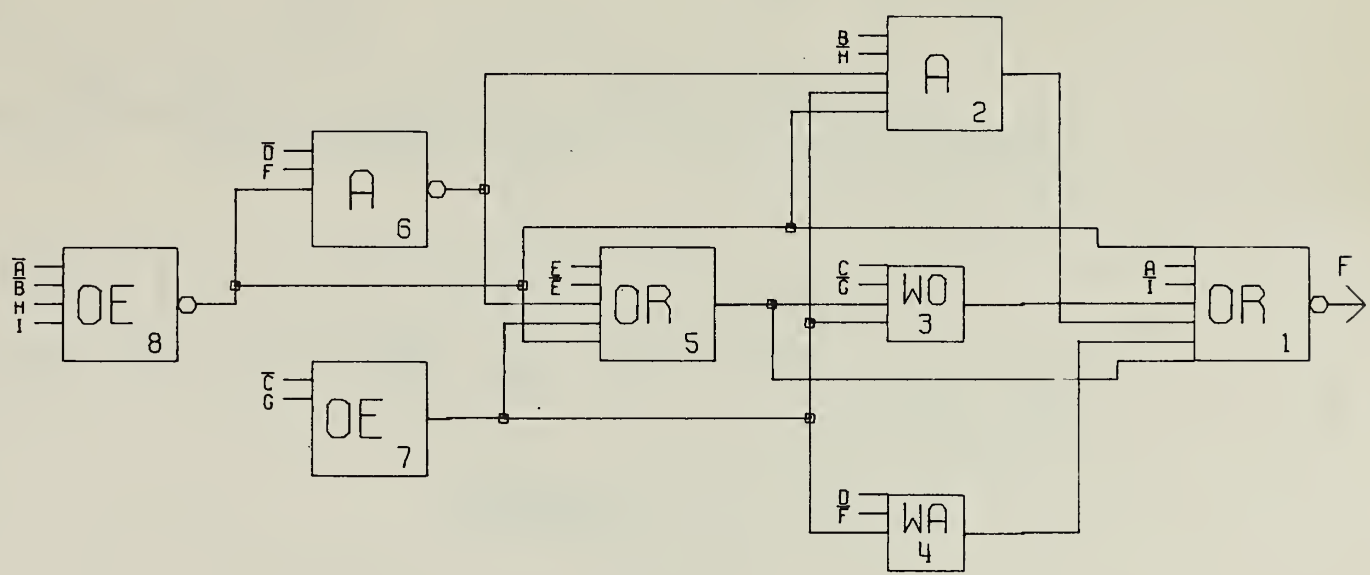
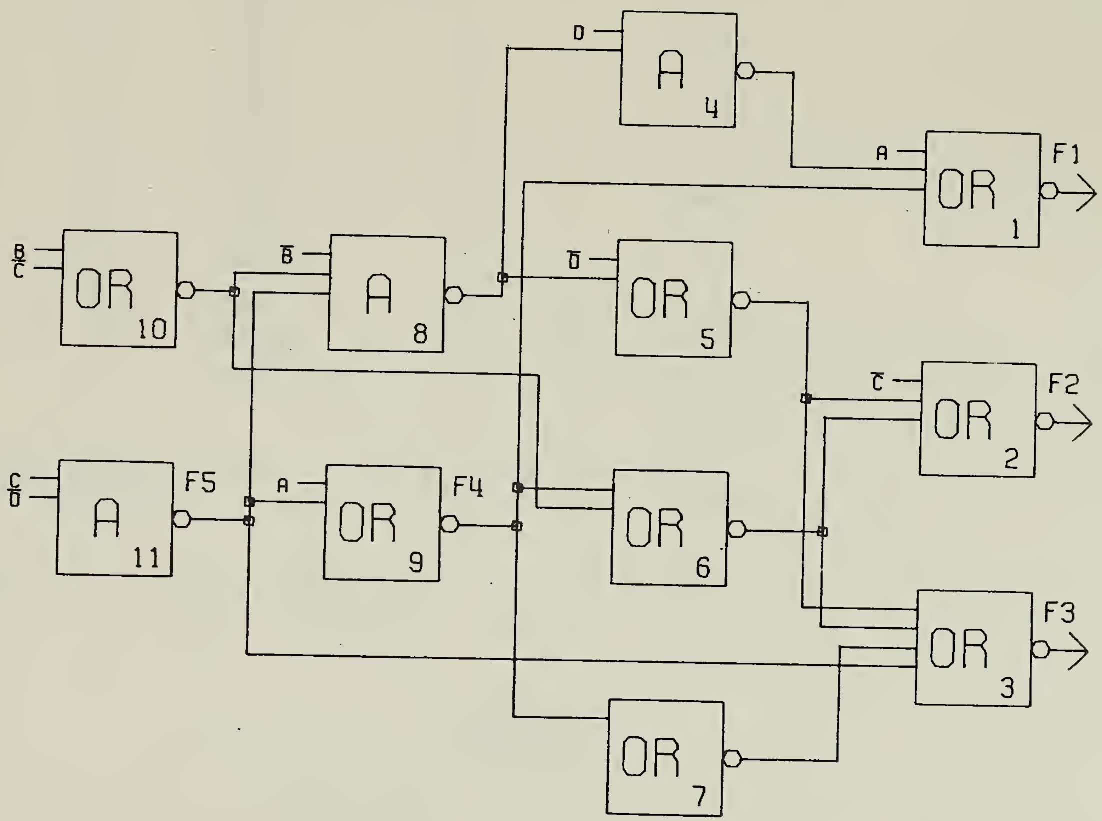


Table A-2

```
THESE ARE THE DATA CARDS FOR FIG. A-2:


FIG. A-2    THIS EXAMPLE SHOWS A MULTIPLE OUTPUT
NETWORK AND ALSO DEMONSTRATES GATE LABELING.
        ..
    F1      1    NØR        1        X1   4   9
    F2      2    NØR        1        Y3   5   6
    F3      3    NØR        1         5   6   7  11
            4   NAND        2        X4   8
            5    NØR        2        Y4   8
            6    NØR        2         9  10
            7    NØR        2         9
            8   NAND        3        Y2  10  11
    F4      9    NØR        3        X1  11
           10    NØR        4        X2  Y3
    F5     11   NAND        4        X3  Y4
END
```



NETWORK AND ALSO DEMONSTRATES GATE LABELING.

Table A-3

```
THESE ARE THE DATA CARDS FOR FIG. A-3:

FIG. A-3    THIS EXAMPLE SHOWS THE RESULTS OF
SPECIFYING MORE THAN 5 GATES PER LEVEL.
           ..
     F     1   NOR      1        2   6   8  11  12  13  16
           2   NOR      2        3   4
           3   NOR      3       X1
           4   NOR      3        5
           5   NOR      4       X2
           6   NOR      2        4   7
           7   NOR      3        5
           8   NOR      2       14
           9   NOR      3       10
          10   NOR      4       X3
          11   NOR      2        9  14  15
          12   NOR      2       10
          13   NOR      2       14  16
          14   NOR      3       X4
          15   NOR      3       X5
          16   NOR      3        5  10
END
```

FIG. A-3   THIS EXAMPLE SHOWS THE RESULTS OF SPECIFYING MORE THAN 5 GATES PER LEVEL.

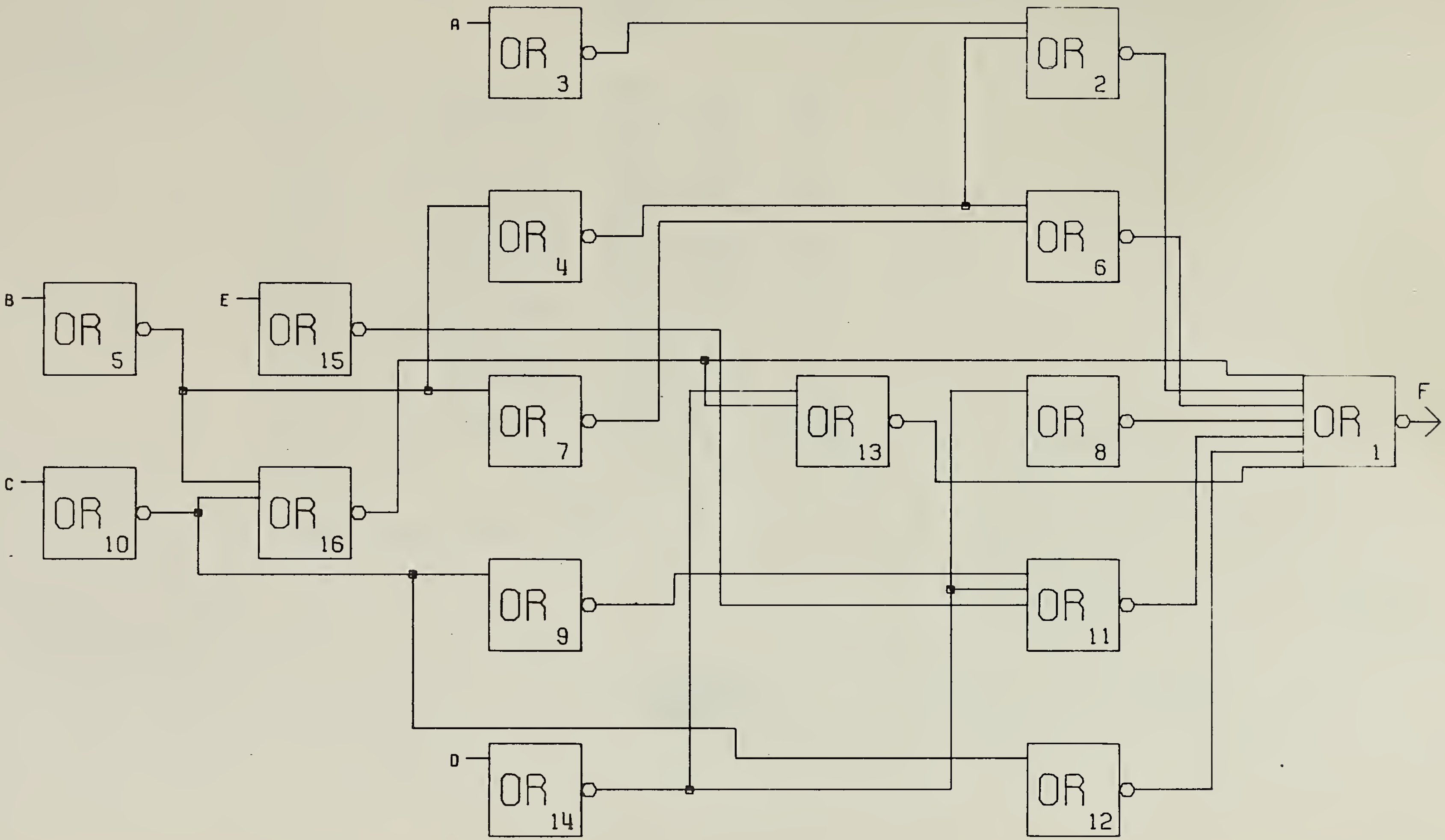


Table A-4

```
THESE ARE THE DATA CARDS FOR FIG. A-4:

FIG. A-4    THIS EXAMPLE SHOWS THE RESULTS OF
USING TOO MANY HEADING CARDS.
THIS IS HEADING CARD NO. 3
THIS IS HEADING CARD NO. 4
THIS IS HEADING CARD NO. 5
THIS IS HEADING CARD NO. 6
THIS IS HEADING CARD NO. 7
THIS IS HEADING CARD NO. 8
THIS IS HEADING CARD NO. 9
THIS IS HEADING CARD NO. 10
THIS IS HEADING CARD NO. 11
            ..
           1    NAND        1          3    4
           2    NAND        3         X2
           3    NAND        2         X1    2
           4    NAND        2          2    5
           5    NAND        3         X3
END
```

FIG. A-4   THIS EXAMPLE SHOWS THE RESULTS OF
USING TOO MANY HEADING CARDS.
THIS IS HEADING CARD NO. 3
THIS IS HEADING CARD NO. 4
THIS IS HEADING CARD NO. 5
THIS IS HEADING CARD NO. 6
THIS IS HEADING CARD NO. 7
ERROR - TOO MANY HEADING CARDS

THIS IS HEADING CARD NO. 9
THIS IS HEADING CARD NO. 10
THIS IS HEADING CARD NO. 11

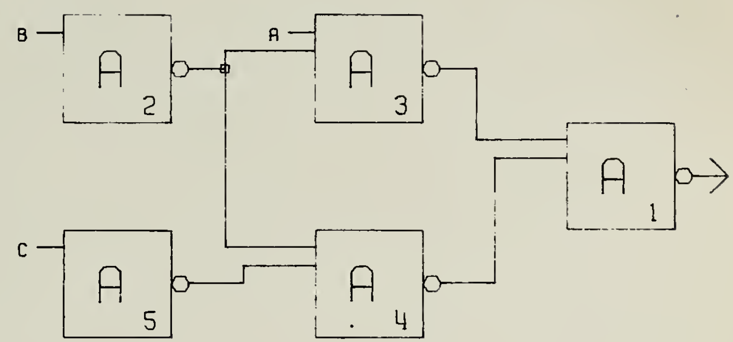


Table A-5

```
THESE ARE THE DATA CARDS FOR FIG. A-5:


FIG. A-5    THIS EXAMPLE SHOWS THE RESULTS OF
SPECIFYING MORE THAN 7 INPUTS TO A SINGLE GATE.
          ..
          1    AND      1       X1  2   3   4   5   6   7   8   9
          2     OR      2       X2  6
          3     OR      2       X3  7
          4     OR      2       X4  8
          5     OR      2       X5  9
          6    AND      3       Y5
          7    AND      3       Y4
          8    AND      3       Y3
          9    AND      3       Y2
END
```

FIG. A-5 THIS EXAMPLE SHOWS THE RESULTS OF SPECIFYING MORE THAN 7 INPUTS TO A SINGLE GATE. FAN-IN OVERFLOW, INPUTS BEYOND 7 IGNORED

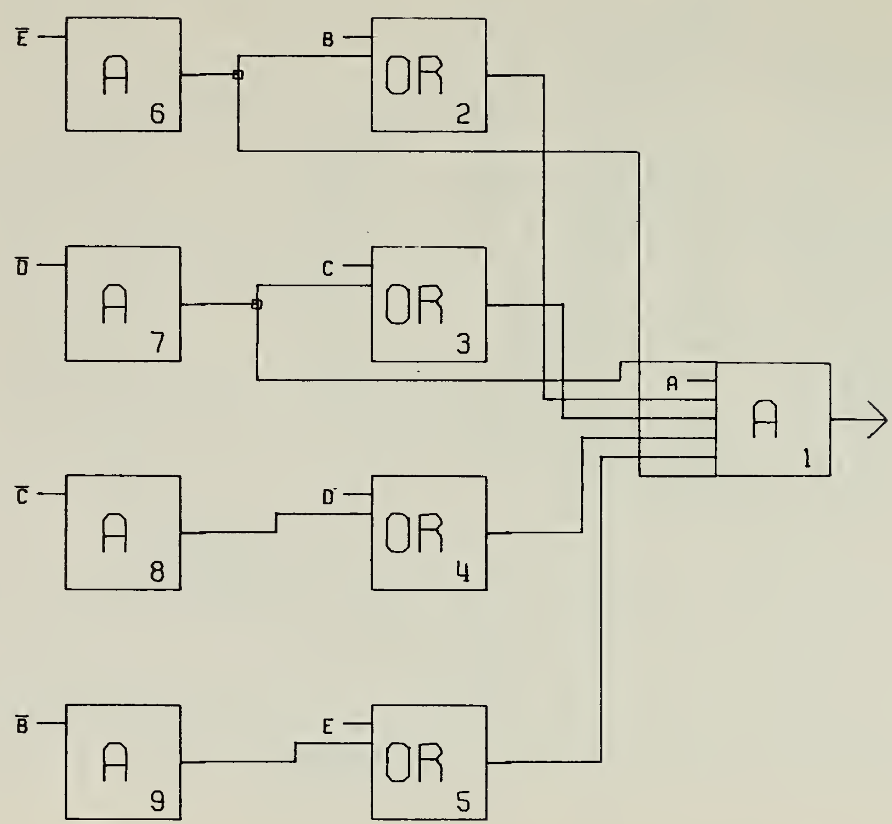


Table A-6

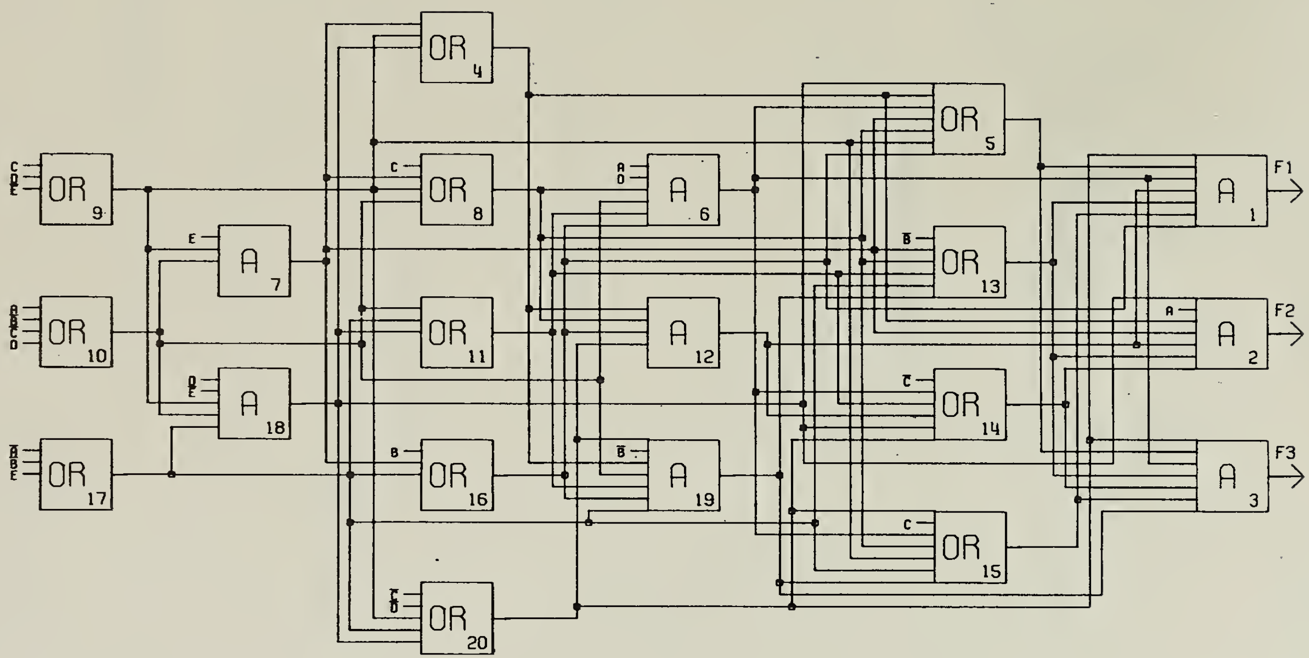
THESE ARE THE DATA CARDS FOR FIG. A-6:

FIG. A-6   THIS IS AN EXAMPLE OF A 20 GATE
NETWORK.

F1	1	AND	1	5	6	15	13	15	12	20
F2	2	AND	1	X1	7	4	14	12	13	18
F3	3	AND	1	6	5	13	14	19	15	20
	4	OR	4	9	7	18				
	5	OR	2	7	18	8	4	9	6	16
	6	AND	3	X1	X4	10	11	8	16	
	7	AND	5	X5	9	10				
	8	OR	4	X3	7	9	10			
	9	OR	6	X3	X4	Y5				
	10	OR	6	X1	Y2	Y3	X4			
	11	OR	4	10	17	18				
	12	AND	3	4	16	8	20			
	13	OR	2	Y2	7	8	17	11	19	
	14	OR	2	Y3	6	12	11	20	18	
	15	OR	2	X3	6	19	20	8	9	17
	16	OR	4	X2	7	17				
	17	OR	6	Y1	Y2	X5				
	18	AND	5	X4	Y5	9	10	17		
	19	AND	3	Y2	4	16	10	11	17	20
	20	OR	4	Y3	Y4	9	18	17		

END

FIG. A-6 THIS IS AN EXAMPLE OF A 20 GATE NETWORK.



APPENDIX B

Program Listing

```
      REAL*4 NOR,NAND
      INTEGER*4 VLBL(20),RVLBL(20),GLEV(20)
      DIMENSION CARD(20),GTYPE(20),          FEDBY(11),NFEDBY(11),NMINLV
     1(20),INTERC(20,20),IEXVAR(18,20),LLRH(60,20),INEXTL(20),
     2 MIDDLE(20),          SPACE(20),MARKGT(20),GATEPS(20,2),NTAKEN(20),
     3NWIRE(20),    LRH(60,20)             ,INPUT(20),            IGTPOS(20)
     4,MARKLV(20),SYM(100,2),ISYM(100),INPTTL(20),          XLEVEL(20)
     5 ,DRAW(1500,2),IDRAW(1500),ZLABEL(20)
      DATA WIREDO,WIRECA/ 'WOR ','WAND'/
      DATA WO,WA,  AND1,AND2 / 'WO'   ,'WA',   ' AND','(AND' /
      DATA NOR,END,TEST,ZAND / ' NOR','END ',' .. ','A'     /
      DATA ZOR,ZNOR,ZX,Z1,Z2 / 'O'    ,'N','  X','  1','  2'/
      DATA ZY,ZU,ZV,NAND     / '  Y' ,'  U' ,'  V','NAND'  /
      DATA ZNAND,XOR,ZXOR    / 'D',' XOR','X'               /
      DATA XNOR,ZXNOR        / 'XNOR','Z'                   /
C**** PRESENT LIMITATIONS: 20 GATES,20 LEVELS,   7 FAN-IN,5   GATES/LEVEL,
C                          5 EXTERNAL VARIABLES
C
C     MOVE PEN AWAY FROM BORDER
C
      CALL CCP1BA
      CALL PLOT(.25,.25,-3)
C
C     READ HEADING CARDS PRECEDING NETWORK DESCRIPTION
C
    1 N=0
      TOP = 9.575
    2 READ 1000,CARD
 1000 FORMAT(20A4)
      N=N+1
      IF(N.GE. 8)GOTO3
      IF(CARD(4).EQ.TEST)GOTO4
      IF(N.GE.2)GOTO5
      CALL SYMBOL(0.,9.575,.15,CARD(1),0.0,50)
      GOTO2
    5 TOP =  TOP - .225
      CALL SYMBOL(.0,TOP,.15,CARD(1),0.0,50)
      GOTO2
C
C     ERROR MESSAGE FOR TOO MANY (.GT. 8) HEADING CARDS
C
    3 TOP =  TOP - .225
      CALL SYMBOL(.0,TOP,.15,'ERROR - TOO MANY HEADING CARDS',0.0,30)
C
C     MOVE ORIGIN AND CONTINUE WRITING
C
      CALL PLOT(8.0,.0,-3)
      GOTO1
C
C     READ CIRCUIT DESCRIPTION NEXT
C
    4 DO 38 I=1,800
   38 IDRAW(I)=0
      NUM=0
      NDRAW=0
      DO 7 I=1,20
      NWIRE(I)=0
      MIDDLE(I)=0
    7 NMINLV(I)=0
      DO 8 I=1,20
      DO 9 J=1,18
```

```
    9 IEXVAR(J,I)=0
      DO 8  J=1,20
    8 INTERC(I,J)=0
      MAXLEV=1
      DO 20 I=1,20
      VLBL(I)=0
      RVLBL(I)=0
      DO 20 J=1,60
      LLRH(J,I)=0
      LRH(J,I)=0
   20 CONTINUE
   94 READ 2000,ATEND,ZLBL ,TYPE,LEVEL,(FEDBY(J),NFEDBY(J),J=1,11)
 2000 FORMAT(A4,A2,10X,  A4,6X,I2,6X,11(    A3,I1))
C
C     IF LAST CARD READ, GO TO SPACE CALCULATION
C
      IF(ATEND.EQ.END)GOTO6
      NUM=NUM+1
      ZLABEL(NUM)=ZLBL
C
C     RECODE GATE TYPE (NOTE: BOTH NOR + OR GATES WILL BE LABELED 'O')
C
      GTYPE(NUM)=ZOR
      IF(TYPE.EQ.XOR)GTYPE(NUM)=ZXOR
      IF(TYPE.EQ.XNOR)GTYPE(NUM)=ZXNOR
      IF(TYPE.EQ.AND1)GTYPE(NUM)=ZAND
      IF(TYPE.EQ.AND2)GTYPE(NUM)=ZAND
      IF(TYPE.EQ.NOR)GTYPE(NUM)=ZNOR
      IF(TYPE.EQ.NAND)GTYPE(NUM)=ZNAND
      IF(TYPE.NE.WIREDA)GOTO83
      NWIRE(NUM)=1
      GTYPE(NUM)=WA
   83 IF(TYPE.NE.WIREDO)GOTO84
      NWIRE(NUM)=1
      GTYPE(NUM)=WO
C
C     STORE LEVEL INFORMATION
C
   84 GLEV(NUM)=LEVEL
      NMINLV(LEVEL)=NMINLV(LEVEL)+1
C     MARK GATES FOR REPOSITIONING IF MORE THAN 5 IN A LEVEL
      IF(NMINLV(LEVEL).LE.5)GOTO143
      GLEV(NUM)=-LEVEL
  143 CONTINUE
C
C     BUILD CONNECTION AND INTERCONNECTION TABLES
C
      IF(NFEDBY(8).EQ.0)GOTO10
      TOP =   TOP - .225
      CALL SYMBOL(.0,TOP,.15,'FAN-IN OVERFLOW, INPUTS BEYOND 7 IGNORED'
     1,0.0,40)
   10 DO 11 I=1,7
      IF(FEDBY(I).EQ.Z1)NFEDBY(I)=NFEDBY(I)+10
      IF(FEDBY(I).EQ.Z2)NFEDBY(I)=20
      IF(NFEDBY(I).EQ.0)GOTO94
      IF(FEDBY(I).NE.ZX.AND.FEDBY(I).NE.ZU)GOTO13
      IEXVAR(NFEDBY(I),NUM)=1
      GOTO11
   13 IF(FEDBY(I).NE.ZY.AND.FEDBY(I).NE.ZV)GOTO12
      IEXVAR(NFEDBY(I)+9,NUM)=1
      GOTO11
```

```
 12 INTERC(NFEDBY(I),NUM)=1
 11 CONTINUE
    GOTO94
C
C     MAKE ADJUSTMENTS FOR LEVELS WITH MORE THAN 5 GATES
C
  6 DO 144 LL=1,20
    DO 145 LG=1,NUM
    IF(GLEV(LG).EQ.-LL)GOTO146
145 CONTINUE
    GOTO144
146 DO 147 LGG=1,NUM
    IF(GLEV(LGG).LT.-LL)GLEV(LGG)=GLEV(LGG)-1
    IF(GLEV(LGG).GT. LL)GLEV(LGG)=GLEV(LGG)+1
    IF(GLEV(LGG).EQ.-LL)GLEV(LGG)=LL+1
147 CONTINUE
144 CONTINUE
C     DETERMINE MAXIMUM NUMBER OF LEVELS NEEDED AND RECALCULATE NMINLV
    MAXLEV=0
    DO 149 I=1,20
149 NMINLV(I)=0
    DO 148 I=1,NUM
    NMINLV(GLEV(I))=NMINLV(GLEV(I))+1
    IF(GLEV(I).GT.MAXLEV)MAXLEV        =GLEV(I)
148 CONTINUE
C
C     CALCULATE SPACING BETWEEN LEVELS
C
    IPTSYM=0
    DO 27 I=1,NUM
    INPUT(I)=1
    DO 17 II=1,18
    IF(IEXVAR(II,I).EQ.1)INPUT(I)=INPUT(I)+1
 17 CONTINUE
    INEXTL(I)=INPUT(I)-1
    INPTTL(I)=INEXTL(I)
    DO 27 II=2,NUM
    IF(INTERC(II,I).EQ.1)INPTTL(I)=INPTTL(I)+1
 27 CONTINUE
C
C     CALCULATE Y-COORDINATES OF GATES
C
    DO 19 I=1,MAXLEV
 19 NTAKEN(I)=0
    DO 18 I=1,NUM
    GATEPS(I,2)=.75*(NMINLV(GLEV(I))-1)-1.5*NTAKEN(GLEV(I))
    IGTPOS(I)=6*(NMINLV(GLEV(I))-1)-12*NTAKEN(GLEV(I))+30
    IRES=IGTPOS(I)-4
    DO 30 IJK=1,7
 30 LLRH(IRES+IJK,GLEV(I))=1
    ABOVE DO LOOP RESERVES SPACE OCCUPIED BY GATES
 18 NTAKEN(GLEV(I))=NTAKEN(GLEV(I))+1
    DO 85 I=1,NUM
    IF(NWIRE(I).EQ.0)GOTO85
    IF(INPTTL(I).LE.5)GOTO85
    KLAG=0
 86 NDRAW=NDRAW+1
    IDRAW(NDRAW)=GLEV(I)
    DRAW(NDRAW,1)=.375
    DRAW(NDRAW,2)=-.25+KLAG*.50
    NDRAW=NDRAW+1
```

```
      IDRAW(NDRAW)=GLEV(I)
      DRAW(NDRAW,1)=.375
      DRAW(NDRAW,2)=-.375+KLAG*.75
      NDRAW=NDRAW+1
      IDRAW(NDRAW)=-1
      IF(INPTTL(I).LE.6)GOTO85
      IF(KLAG.EQ.1)GOTO85
      KLAG=1
      GOTO86
   85 CONTINUE
C     CHECK EACH GATE FOR INPUT LINE CONFLICTS AT POSITION 3
      DO 68 J=1,NUM
      IF(INEXTL(J).GE.3)GOTO68
      KCOUNT=INEXTL(J)
      DO 69 K=1,NUM
      IF(INTERC(K,J).EQ.1)KCOUNT=KCOUNT+1
      IF(KCOUNT.EQ.3)GOTO70
   69 CONTINUE
      GOTO68
   70 IF(GLEV(K).NE.GLEV(J)+1)GOTO68
      IF(IGTPOS(K).LE.IGTPOS(J))GOTO68
      DO 80 KK=1,NUM
      IF(GLEV(KK).NE.GLEV(J)+1.OR.IGTPOS(KK).NE.IGTPOS(J))GOTO80
      MIDDLE(J)=1
C     ABOVE STATEMENT MARKS GATE J AS HAVING AN INPUT CONFLICT AT POS. 3
   80 CONTINUE
   68 CONTINUE
C
C     CALCULATE X-COORDINATES OF GATES
C
      DO 41 II=1,NUM
      IF(INEXTL(II).EQ.INPTTL(II))GOTO41
      INPTTL(II)=INPTTL(II)+MIDDLE(II)
      IA=INPTTL(II)
      IF(IA.GE.7)IA=6
      IA=IA-INEXTL(II)
      IB=IGTPOS(II)+3-INEXTL(II)
      DO 42 IJ=1,IA
   42 LRH(IB-IJ,GLEV(II))=1
      IF(INPTTL(II).NE.7)GOTO41
      LRH(IGTPOS(II)+3,GLEV(II))=1
   41 CONTINUE
      DO 28 I=2,NUM
      DO 29 II=1,MAXLEV
   29 MARKLV(II)=0
      IFLAG=0
      LEVFT=-1
      DO 93 J=1,NUM
   93 IF(INTERC(I,J).EQ.1)MARKLV(GLEV(J))=MARKLV(GLEV(J))+1
      DO 32 J=1,MAXLEV
      IF(MARKLV(J).NE.0)GOTO34
   32 CONTINUE
      GOTO28
C        J = RIGHTMOST LEVEL FED BY GATE I
   34 IF(J+1.EQ.GLEV(I))IFLAG=1
      IF(MARKLV(J).EQ.1.AND.IFLAG.EQ.0)GOTO35
C        RESERVE A VERTICAL LINE AT THIS LEVEL
      IF(IFLAG.EQ.1)GOTO46
      LINEV=VLBL(J)+1
      VLBL(J)=VLBL(J)+1
      GOTO47
```

```
46 LINEV=RVLBL(J)+1
   RVLBL(J)=RVLBL(J)+1
C     FIND MAX AND MIN Y-COORDS
47 MAX=0
   MIN=60
   ICOUNT=0
   DO 36 JJ=1,NUM
   IF(INTERC(I,JJ).EQ.0)GOTO36
   IF(GLEV(JJ).NE.J)GOTO36
   JJLEV=IGTPOS(JJ)+3-INPUT(JJ)
   IF(MIDDLE(JJ).EQ.0)GOTO71
   IF(INPUT(JJ).LT.3)GOTO  71
   IF(INPUT(JJ).GT.3)GOTO72
   IF(IGTPOS(I).LE.IGTPOS(JJ))GOTO 71
   JJLEV=JJLEV-1
   GOTO71
72 IF(IGTPOS(I).GT.IGTPOS(JJ))GOTO 71
   JJLEV=IGTPOS(JJ)
71 IF(JJLEV     .GT.MAX)MAX=JJLEV
   IF(JJLEV     .LT.MIN)MIN=JJLEV
   ICOUNT=ICOUNT+1
36 CONTINUE
   IF(ICOUNT.EQ.MARKLV(J))GOTO37
   IF(LEVFT.LT.0)GOTO37
   IF(LEVFT.GT.MAX)MAX=LEVFT
   IF(LEVFT.LT.MIN)MIN=LEVFT
37 CONTINUE
   IF(IFLAG.EQ.0)GOTO49
   ISECT=0
C  SET A FLAG 'KDIRFD' FOR THE SPECIAL CASE OF A SINGLE INPUT TO THIS
C  LEVEL DIRECTLY (I.E. NO VERTICAL JOGS) FED BY A GATE IN THE NEXT
C  LEVEL
   KDIRFD=0
   IF(IGTPOS(I).EQ.MAX.AND.MAX.EQ.MIN.AND.IFLAG.EQ.1)KDIRFD=1
   IF(IGTPOS(I).EQ.MAX.OR.IGTPOS(I).EQ.MIN)ISECT=1
   IF(IGTPOS(I).GT.MAX)MAX=IGTPOS(I)
   IF(IGTPOS(I).LT.MIN)MIN=IGTPOS(I)
   IF(KDIRFD.EQ.0)GOTO49
   ISECT=0
C  UNRESERVE VERTICAL LINE
   RVLBL(J)=RVLBL(J)-1
   LINEV=RVLBL(J)
   X=-.75-LINEV*.125
   GOTO97
C
C  STORE INFORMATION FOR DRAWING VERTICAL LINE
C
49 NDRAW=NDRAW+1
   IDRAW(NDRAW)=J+IFLAG
   Y=(MAX-30)*.125
   X=.75+LINEV*.125
   IF(IFLAG.EQ.1)X=-.75-(LINEV)*.125+.125
   DRAW(NDRAW,1)=X
   DRAW(NDRAW,2)=Y
   NDRAW=NDRAW+1
   IDRAW(NDRAW)=J+IFLAG
   Y=(MIN-30)*.125
   DRAW(NDRAW,1)=X
   DRAW(NDRAW,2)=Y
   NDRAW=NDRAW+1
   IDRAW(NDRAW)=-1
```

```
C
C     RESERVE AND DRAW (W/ CONNECTION PTS) HORIZONTAL LINES FROM PREVIOUS
C     VERTICAL LINE TO GATES BEING FED
C
   97 ICOUNT=0
      DO 39 JJ=1,NUM
      IF(GLEV(JJ).NE.J)GOTO39
      IF(INTERC(I,JJ).EQ.0)GOTO39
      JJLEV=IGTPOS(JJ)+3-INPUT(JJ)
      IF(MIDDLE(JJ).EQ.0)GOTO74
      IF(INPUT(JJ).LT.3)GOTO74
      IF(INPUT(JJ).GT.3)GOTO75
      IF(IGTPOS(I).LE.IGTPOS(JJ))GOTO76
      JJLEV=JJLEV-1
      INPUT(JJ)=INPUT(JJ)+1
      GOTO74
   76 MIDDLE(JJ)=0
      GOTO74
   75 IF(IGTPOS(I).GT.IGTPOS(JJ))GOTO74
      JJLEV=IGTPOS(JJ)
      MIDDLE(JJ)=0
      GOTO77
   74 INPUT(JJ)=INPUT(JJ)+1
   77 IF(IFLAG.EQ.0)LRH(JJLEV,J)=0
      IF(INPUT(JJ).GE.7)INPUT(JJ)=0
      NDRAW=NDRAW+1
      IDRAW(NDRAW)=J
      Y=(JJLEV-30)*.125
      DRAW(NDRAW,1)=.375
      DRAW(NDRAW,2)=Y
      NDRAW=NDRAW+1
      IDRAW(NDRAW)=J+IFLAG
      DRAW(NDRAW,1)=X
      DRAW(NDRAW,2)=Y
      NDRAW=NDRAW+1
      IDRAW(NDRAW)=-1
      IF(JJLEV.EQ.MAX)GOTO39
      IF(JJLEV.EQ.MIN)GOTO39
C
C     DRAW INTERCONNECTION POINTS
C
      IPTSYM=IPTSYM+1
      ISYM(IPTSYM)=J+IFLAG
      SYM(IPTSYM,1)=X
      SYM(IPTSYM,2)=Y
   39 CONTINUE
      IF(LEVFT.LT.0)GOTO40
      IF(LEVFT.EQ.MAX)GOTO40
      IF(LEVFT.EQ.MIN)GOTO40
      IPTSYM=IPTSYM+1
      ISYM(IPTSYM)=J+IFLAG
      Y=(LEVFT-30)*.125
      SYM(IPTSYM,1)=X
      SYM(IPTSYM,2)=Y
      GOTO40
   35 IF(LEVFT.LT.0)GOTO95
      MAX=LEVFT
      MIN=MAX
      X=.75+VLBL(J)*.125+.125
      GOTO40
   95 DO 96 JJ=1,NUM
```

```
      IF(GLEV(JJ).NE.J)GOTO96
      IF(INTERC(I,JJ).EQ.0)GOTO96
      MAX=IGTPOS(JJ)+3-INPUT(JJ)
      IF(MIDDLE(JJ).EQ.0)GOTO78
      IF(INPUT(JJ).LT.3)GOTO78
      IF(INPUT(JJ).GT.3)GOTO79
      IF(IGTPOS(I).LE.IGTPOS(JJ))GOTO78
      MAX=MAX-1
      GOTO78
   79 IF(IGTPOS(I).GT.IGTPOS(JJ))GOTO78
      MAX=IGTPOS(JJ)
   78 MIN=MAX
      X=.75+VLBL(J)*.125+.125
      GOTO97
   96 CONTINUE
C
C     DOES NEXT LEVEL CONTAIN GATE I ?
C
   40 IF(IFLAG.EQ.0)GOTO48
C
C     CONNECT INTERCONNECTION TREE TO FEEDING GATE
C
      NDRAW=NDRAW+1
      IDRAW(NDRAW)=J+1
      Y=(IGTPOS(I)-30)*.125
      DRAW(NDRAW,1)=X
      DRAW(NDRAW,2)=Y
      NDRAW=NDRAW+1
      IDRAW(NDRAW)=J+1
      DRAW(NDRAW,1)=-.5
      DRAW(NDRAW,2)=Y
      NDRAW=NDRAW+1
      IDRAW(NDRAW)=-1
      IF(IGTPOS(I).EQ.MIN.AND.ISECT.EQ.0)GOTO28
      IF(IGTPOS(I).EQ.MAX.AND.ISECT.EQ.0)GOTO28
      IPTSYM=IPTSYM+1
      ISYM(IPTSYM)=J+1
      SYM(IPTSYM,1)=X
      SYM(IPTSYM,2)=Y
      GOTO28

     CAN A LINE BETWEEN MAX AND MIN,AT THE SAME HEIGHT AS GATE I, BE
     FED THROUGH TO THE NEXT LEVEL LEFT ? (NOTE IFLAG=0 IN THIS SECTION)

   48 IF(IGTPOS(I).GT.MAX.OR.IGTPOS(I).LT.MIN)GOTO43
      IF(LLRH(IGTPOS(I),J+1).NE.0)GOTO43
      IF(LRH(IGTPOS(I),J).NE.0)GOTO43
      IF(LRH(IGTPOS(I),J+1).NE.0)GOTO43
      ICHOSE=IGTPOS(I)
      INTRSC=1
      IF(MAX.EQ.MIN)INTRSC=0
      GOTO44

     IF NOT, CAN A LINE BETWEEN CURRENT MAX AND MIN, CLOSE TO THE HEIGHT
     OF GATE I, S  FED THRU TO THE NEXT LEVEL LEFT (NOTE IFLAG=0) ?

   43 ICHOSE=-1
      DO 51 LVL=MIN,MAX
      IF(LLRH(LVL,J+1).NE.0)GOTO51
      IF(LRH(LVL,J).NE.0)GOTO51
      IF(LRH(LVL,J+1).NE.0)GOTO51
```

```
      IF(IABS(LVL-IGTPOS(I)).GE.IABS(ICHOSE-IGTPOS(I)))GOTO51
      IF(J+2.NE.GLEV(I))GOTO99
      IF(LVL.GE.IGTPOS(I))GOTO99
      JPLUS2=J+2
      DO 98 IX=2,NUM
      IF(GLEV(IX).NE.JPLUS2)GOTO98
      IF(LVL.NE.IGTPOS(IX))GOTO98
      GOTO51
   98 CONTINUE
   99 ICHOSE=LVL
   51 CONTINUE
      IF(ICHOSE.EQ.-1)GOTO50
      INTRSC=1
      IF(MAX.EQ.MIN)INTRSC=0
      GOTO44
C
C     IF FEED-THRU LINE STILL NOT FOUND, VERTICAL LINE MUST BE EXTENDED
C     TO MEET THE CLOSEST AVAILABLE FEED-THRU LINE
C
   50 IF(MAX.NE.MIN)GOTO52
      LINEV=VLBL(J)+1
      VLBL(J)=VLBL(J)+1
      X=.75+LINEV*.125
   52 ICHOSE=-1
      ICENTR=(MIN+MAX)/2
      DO 53 LVL=1,60
      IF(LLRH(LVL,J+1).NE.0)GOTO53
      IF(LRH(LVL,J).NE.0)GOTO53
      IF(LRH(LVL,J+1).NE.0)GOTO53
      IF(GLEV(I).GE.J+3)GOTO102
      IF(IABS(LVL-IGTPOS(I)).GE.IABS(ICHOSE-IGTPOS(I)))GOTO53
      GOTO103
  102 IF(IABS(LVL-IGTPOS(I))+2*IABS(LVL-ICENTR).GE.
     1   IABS(ICHOSE-IGTPOS(I))+2*IABS(ICHOSE-ICENTR))GOTO53
  103 IF(J+2.NE.GLEV(I))GOTO101
      IF(LVL.GE.IGTPOS(I))GOTO101
      JPLUS2=J+2
      DO 100 IX=2,NUM
      IF(GLEV(IX).NE.JPLUS2)GOTO100
      IF(LVL.NE.IGTPOS(IX))GOTO100
      GOTO53
  100 CONTINUE
  101 ICHOSE=LVL
   53 CONTINUE
      Y=(MIN-30)*.125
      IF(ICHOSE.GT.MAX)Y=(MAX-30)*.125
      NDRAW=NDRAW+1
      IDRAW(NDRAW)=J
      DRAW(NDRAW,1)=X
      DRAW(NDRAW,2)=Y
      NDRAW=NDRAW+1
      IDRAW(NDRAW)=J
      DRAW(NDRAW,1)=X
      DRAW(NDRAW,2)=(ICHOSE-30)*.125
      NDRAW=NDRAW+1
      IDRAW(NDRAW)=-1
      IF(MIN.EQ.MAX)GOTO54
      IPTSYM=IPTSYM+1
      ISYM(IPTSYM)=J
      SYM(IPTSYM,1)=X
      SYM(IPTSYM,2)=Y
```

```
   54 INTRSC=0
      GOTO44
C
C     DRAW FEED-THRU LINE AND DO ASSOCIATED BOOKKEEPPING (ICHOSE=FEED-
C     THRU LINE)
C
   44 MARKLV(J+1)=MARKLV(J+1)+1
      LLRH(ICHOSE,J+1)=1
      LRH(ICHOSE,J)=1
      IF(J+2.EQ.GLEV(I))LRH(ICHOSE,J+1)=1
      LEVFT=ICHOSE
      NDRAW=NDRAW+1
      IDRAW(NDRAW)=J
      DRAW(NDRAW,1)=X
      DRAW(NDRAW,2)=(ICHOSE-30)*.125
      NDRAW=NDRAW+1
      IF(GLEV(I).EQ.J+2)GOTO55
      IDRAW(NDRAW)=J+1
      DRAW(NDRAW,1)=.75+VLBL(J+1)*.125+.125
      GOTO56
   55 IDRAW(NDRAW)=J+2
      DRAW(NDRAW,1)=-.75-RVLBL(J+1)*.125
   56 DRAW(NDRAW,2)=(ICHOSE-30)*.125
      NDRAW=NDRAW+1
      IDRAW(NDRAW)=-1
      IF(INTRSC.EQ.0)GOTO57
      IPTSYM=IPTSYM+1
      ISYM(IPTSYM)=J
      SYM(IPTSYM,1)=X
      SYM(IPTSYM,2)=(ICHOSE-30)*.125
   57 J=J+1
      GOTO34
   28 CONTINUE
C
C     COMPUTE X-COORD OF EACH LEVEL
C
      XLEVEL(1)=0.0
      DO 58 I=2,MAXLEV
   58 XLEVEL(I)=XLEVEL(I-1)+1.500+(VLBL(I-1)+RVLBL(I-1))*.125
C
C     MOVE PEN AND ORIGIN TO CENTER OF OUTPUT GATE
C
      XMAX=1.+XLEVEL(MAXLEV)
      CALL PLOT(XMAX,3.75,-3)
C
C     DRAW GATES
C
      DO 59 I=1,NUM
      XCOORD=-XLEVEL(GLEV(I))
      YCOORD=GATEPS(I,2)
      TYP=GTYPE(I)
      ZLBL=ZLABEL(I)
      IF(NWIRE(I).EQ.1)GOTO82
      CALL GATE(XCOORD,YCOORD,I,TYP,ZLBL)
      GOTO59
   82 CALL WIRED(XCOORD,YCOORD,I,TYP,ZLBL)
   59 CONTINUE
C
C     DRAW ARROW
C
      KK=NMINLV(1)
```

```
      YINTL=-3.0*(KK-1)/4.0
      DO 104 J=1,KK
      Y=YINTL+(J-1)*1.5
      CALL PLOT(.500, Y,3)
      CALL PLOT(.750, Y,2)
      YY=Y+.125
      CALL PLOT(.625,YY,2)
      YY=Y-.125
      CALL PLOT(.625,YY,3)
      CALL PLOT(.750, Y,2)
  104 CONTINUE
C
C     DRAW INTERCONNECTIONS
C
      I=0
   61 IF(I.GE.NDRAW)GOTO62
      I=I+1
      IPEN=3
   60 XCOORD=-DRAW(I,1)-XLEVEL(IDRAW(I))
      YCOORD=DRAW(I,2)
      CALL PLOT(XCOORD,YCOORD,IPEN)
      I=I+1
      IF(IDRAW(I).LT.0)GOTO61
      IPEN=2
      GOTO60
C
C     DRAW INTERCONNECTION POINTS
C
62    IF(IPTSYM.EQ.0)GOTO81
      DO 63 I=1,IPTSYM
      XCOORD=-SYM(I,1)-XLEVEL(ISYM(I))
      YCOORD=SYM(I,2)
      CALL SYMBOL(XCOORD,YCOORD,.06,0,0.0,-1)
   63 CONTINUE
C
C     DRAW EXTERNAL INPUTS
C
   81 DO 65 I=1,NUM
      XCENTR=-XLEVEL(GLEV(I))
      YCENTR=GATEPS(I,2)+.25
      K=0
      DO 65 II=1,9
      DO 65 III=1,2
      J=II+9*III-9
      IF(IEXVAR(J,I).EQ.0)GOTO65
      L=J
      IF(J.GT.9)L=J-9
      LETTER=64+L
C     DRAW INPUT LINE
      X=XCENTR-.375
      Y=YCENTR-K*.125
      K=K+1
      CALL PLOT(X,Y,3)
      X=X-.185
      CALL PLOT(X,Y,2)
      Y=Y-.060
      X=X-.125
C     DRAW BAR, IF ONE IS NEEDED
      IF(J.LE.9)GOTO67
      YPRIME = Y + .005
      CALL SYMBOL(X,YPRIME,.09,26,0.0,-1)
```

```
C      DRAW SYMBOL
   67 CALL SYMBOL(X,Y,.09,LETTER,0.0,-1)
   65 CONTINUE
C
C      SHIFT PEN AND ORIGIN FOR NEXT NETWORK
C
      ZMOST=6.0
      IF(6.0.LT.XMAX+.75)ZMOST=XMAX+.75
      ZMOVE=ZMOST-XMAX-.75+1.5
   64 CALL PLOT(ZMOVE,-3.75,-3)
C
C      RETURN TO READ NEXT CIRCUIT DESCRIPTION
C
      CALL CCP1BA
      GOTO1
      END
```

```
      SUBROUTINE GATE(P,Q,NUMBER,TYPE,ZLABEL)
      DIMENSION ANUM(20)
      DATA ANUM /'1','2','3','4','5','6','7','8','9','10','11','12',
     1      '13','14','15','16','17','18','19','20'/
      DATA ZN,ZO/'N','O'/
      DATA BB / '  ' /
      DATA ZD,ZOR,ZA,ZX,ZOE,ZZ/ 'D','OR','A','X','OE','Z'  /
      NEGATE=0
      ADJ=0.
      ICHAR=1
      DO 5 I=1,20
      IF(I.EQ.NUMBER)DIGIT=ANUM(I)
5     CONTINUE
      IF(NUMBER.GE.10) ADJ=.065
      IF(NUMBER.GE.10) ICHAR=2
C
C     SET OFFSET SO PLOTS WILL CENTER ON GATE CENTER (P,Q)
C
      XOFF=-P
      YOFF=-Q
      CALL OFFSET(XOFF,1.0,YOFF,1.0)
C
C     WRITE SYMBOLS IDENTIFYING GATE TYPE AND NUMBER
C
      IF(TYPE.NE.ZZ)GOTO9
      TYPE=ZX
      NEGATE=1
      GOTO4
    9 IF(TYPE.NE.ZD)GOTO2
      NEGATE=1
      TYPE=ZA
    2 IF(TYPE.NE.ZN)GOTO6
      TYPE=ZO
      NEGATE=1
    6 IF(TYPE.NE.ZO)GOTO4
      XSYM=P-.275
      YSYM=Q-.125
      CALL SYMBOL(XSYM,YSYM,.25,ZOR,0.0,2)
      GOTO3
    4 IF(TYPE.NE.ZX)GOTO1
      XSYM=P-.275
      YSYM=Q-.125
      CALL SYMBOL(XSYM,YSYM,.25,ZOE,0.0,2)
      GOTO3
    1 XSYM=P-.125
      YSYM=Q-.125
      CALL SYMBOL(XSYM,YSYM,.25, ZA ,0.0, 1)
    3 XSYM=P+.19-ADJ
      YSYM=Q-.315
      CALL SYMBOL(XSYM,YSYM,.125,DIGIT,0.0,ICHAR)
      IF(ZLABEL.EQ.BB)GOTO10
      XSYM=P+.470
      YSYM=Q+.185
      CALL SYMBOL(XSYM,YSYM,.125,ZLABEL,0.0,2)
   10 CONTINUE
C
C     DRAW NOR GATE
C
      CALL PLOT(.375,-.375,13)
      CALL PLOT(.375,.0,12)
      IF(NEGATE.EQ.0)GOTO7
```

```
      CALL PLOT(.405,.055,12)
      CALL PLOT(.470,.055,12)
      CALL PLOT(.500,.0,12)
      CALL PLOT(.470,-.055,12)
      CALL PLOT(.405,-.055,12)
      GOTO8
7     CALL PLOT(.540,.0,12)
8     CALL PLOT(.375,.0,12)
      CALL PLOT(.375,.375,12)
      CALL PLOT(-.375,.375,12)
      CALL PLOT(-.375,-.375,12)
      CALL PLOT(.375,-.375, 12)
      RETURN
      END
```

```
      SUBROUTINE WIRED(P,Q,NUMBER,TYPE,ZLABEL)
      DIMENSION ANUM(20)
      DATA ANUM /'1','2','3','4','5','6','7','8','9','10','11','12',
     1     '13','14','15','16','17','18','19','20' /
      DATA BB / '  ' /
      ADJ=0.
      ICHAR=1
      DO 1 I=1,20
      IF(I.EQ.NUMBER)DIGIT=ANUM(I)
    1 CONTINUE
      IF(NUMBER.LT.10)GOTO2
      ADJ=.055
      ICHAR=2
C
C     SET OFFSET SO PLOTS WILL CENTER ON POINT (P,Q)
C
    2 XOFF=-P
      YOFF=-Q
      CALL OFFSET(XOFF,1.0,YOFF,1.0)
C
C     WRITE GATE TYPE AND NUMBER
C
      XSYM=P-.266
      YSYM=Q-0.0
      CALL SYMBOL(XSYM,YSYM,.188,TYPE,0.0,2)
      XSYM=-.164-ADJ+P
      YSYM=Q-.188
      CALL SYMBOL(XSYM,YSYM,.125,DIGIT,0.0,ICHAR)
      IF(ZLABEL.EQ.BB)GOTO10
      XSYM=P+.345
      YSYM=Q+.125
      CALL SYMBOL(XSYM,YSYM,.125,ZLABEL,0.0,2)
   10 CONTINUE
C
C     DRAW WIRED GATE
C
      CALL PLOT(.125,-.02,13)
      CALL PLOT(.125, .25,12)
      CALL PLOT(-.375,.25,12)
      CALL PLOT(-.375,-.25,12)
      CALL PLOT(.125,-.25,12)
      CALL PLOT(.125,.00,12)
      CALL PLOT(.52,.00,12)
      RETURN
      END
```

BLIOGRAPHIC DATA EET	1. Report No. UIUCDCS-R-73-580	2.	3. Recipient's Accession No.

Title and Subtitle

USE AND DESCRIPTION OF THE CALCOMP PROGRAM TO DRAW NETWORKS OF LOGIC GATES

5. Report Date
June, 1973

6.

Author(s)
Jay Culliney

8. Performing Organization Rept. No. UIUCDCS-R-73-580

Performing Organization Name and Address

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois

10. Project/Task/Work Unit No.

11. Contract/Grant No.
NSF GJ-503

Sponsoring Organization Name and Address

National Science Foundation
1800 G Street, N.W.
Wasington, D.C. 20550

13. Type of Report & Period Covered
Technical

14.

Supplementary Notes

Abstracts

A program for automatically drawing networks of logic gates is presented. Also details necessary for the use of the program are explained, and guidelines for modification or extension of the program are given. The program is written in FORTRAN and utilizes a CALCOMP plotter to produce the drawing.

Key Words and Document Analysis. 17a. Descriptors

Logic design, logic circuits, logical elements, programs (computers), plotters.

Identifiers/Open-Ended Terms

Feed-forward networks, computer-aided-design, American Standard Graphic Symbols, Uniform Shapes.

COSATI Field/Group

Availability Statement	19. Security Class (This Report)	21. No. of Pages
Release unlimited	UNCLASSIFIED	47
	20. Security Class (This Page) UNCLASSIFIED	22. Price